IN THE CLAIMS

Please amend the claims to the following.

1. (Currently Amended) An apparatus comprising:

a processor including:

marking logic to mark instruction information for an instruction of a speculative thread as speculative; and

blocker logic to prevent data associated with a speculative store instruction of the speculative thread from being forwarded to a[[n]] non-speculative instruction of a non-speculative thread based on information associated with the speculative store instruction of the speculative thread and information associated with the non-speculative instruction of the non-speculative thread~~, the blocker logic further to prevent the data from being stored in a memory system~~.

2. (Original) The apparatus of claim 1, wherein: blocker logic is further to allow the data associated with a store instruction of the speculative thread to be forwarded to an instruction of a second speculative thread.

3. (Currently Amended) The apparatus of claim 1, further comprising: a ~~plurality of~~ store request buffer[[s]], a ~~each~~ store request buffer entry of the store request buffer including a speculation identifier field to hold a speculation identifier to indicate the store instruction is speculative and an address field to hold an address associated with the store instruction, wherein the information associated with the store instruction includes the speculation identifier and the address.

4. (Currently Amended) The apparatus of claim 3 [[1]], wherein the information associated with the non-speculative instruction of the non-speculative thread includes a second speculation identifier to indicate the non-speculative instruction is non-speculative thread and a second address associated with the non-speculative instruction, and wherein blocker logic to prevent data associated with the speculative store instruction from being forwarded to the non-speculative instruction based on the speculation identifier, the address, the second address, and the second speculation identifier comprises: blocker logic to prevent data associated with the speculative store instruction from being forwarded to the non-speculative instruction in response to a comparison of a first combination of the speculative identifier and the address not matching a second combination of the second speculative identifier and the second address. ~~the memory system further comprises: a data cache that includes a safe-store indicator field associated with each entry of a tag array~~.

5. (Cancelled)

6. (Currently) The apparatus of claim 1, wherein blocker logic further includes:

dependence blocker logic to prevent data associated with a speculative store instruction from being forwarded to an instruction of the non-speculative thread based on information associated with the store instruction of the speculative thread and information associated with the non-speculative thread; and

store blocker logic to prevent the data from being stored in a memory system.

7. (Previously Amended) The apparatus of claim 6, wherein: store blocker logic is outside an execution pipeline.

8. (Cancelled)

9. (Original) The apparatus of claim 6, wherein: dependence blocker logic is included in an execution pipeline.

10. (Previously Amended) The apparatus of claim 9, wherein: dependence blocker logic is included in a memory ordering buffer of the processor.

11. (Currently Amended) A system, comprising:

a memory system that includes a memory device; and

a processor associated with the memory system, the processor including dependence blocker logic to prevent data associated with a store instruction of a speculative thread from being forwarded to an instruction of a non-speculative thread without replacement of the instruction of the non-speculative thread, and to allow the data associated with the store instruction of the speculative thread to be forwarded to a speculative instruction of another speculative thread.

12. (Previously Amended) The system of claim 11, wherein: the processor further includes store blocker logic to prevent the data from being stored in the memory system and marking logic to mark instruction information associated with the store instruction as speculative.

13. (Original) The system of claim 12, wherein: the marking logic is further to associate a safe speculation domain ID with the instruction information.

14. (Original) The system of claim 13, wherein: the marking logic is further to indicate a thread identifier as the speculation domain ID.

15. (Original) The system of claim 12, further comprising: a store request buffer to store the speculation domain ID.

16. (Original) The system of claim 11, wherein: the processor includes a first logical processor to execute the non-speculative thread; and the processor includes a second logical processor to execute the speculative thread.

17. (Original) The system of claim 11, further comprising:

a second processor that includes said dependence blocker logic and said store blocker logic;

wherein said processor is to execute the non-speculative thread and said second processor is to execute the speculative thread.

18. (Previously Amended) The system of claim of claim 11, wherein: the memory system includes store blocker logic to prevent the data from being stored in the memory system and a cache organized to include a plurality of tag lines, wherein each tag line of the cache includes a unique helper thread ID field.

19. (Original) The system of claim 11, wherein: the memory system includes a cache organized to include a plurality of tag lines, wherein each tag line of the cache includes a safe-store indicator field.

20. (Original) The system of claim 11, wherein: the memory system includes a victim tag cache to indicate evicted cache lines that include speculative load data.

21. (Currently) A method, comprising:

receiving instruction information for a load instruction, the instruction information including a load address;

performing a dependence check, wherein performing the dependence check includes:

determining if a store address of an in-flight store instruction matches the load address; and

determining if the load instruction and the in-flight store instruction each originate with a speculative thread;

determining the dependence check is successful in response to determining the store address matches the load address and determining the load instruction and the in-flight store instruction each originate with a speculative thread;

forwarding, if the dependence check is successful, store data associated with the in-flight store instruction to the load instruction; and

declining to forward, if the dependence check is not successful, the store data to the load instruction.

22. (Original) The method of claim 21, wherein performing the dependence check further comprises: determining if the in-flight store instruction and the load instruction originate from the same thread.

23. (Original) The method of claim 22, wherein determining if the in-flight store instruction and the load instruction originate from the same thread further comprises: determining if a thread ID associated with the in-flight store instruction matches a thread ID associated with the load instruction.

24. (Original) The method of claim 21, wherein performing the dependence check further comprises: if the load instruction and the in-flight store instruction do not each originate with a speculative thread, determining if the load instruction and the in-flight store instruction each originate with a non-speculative thread.

25. (Original) The method of claim 21, further wherein: declining to forward further comprises declining to forward the store data to the load instruction if (the load instruction and the in-flight store instruction each originate with a speculative thread) AND (the in-flight store instruction originates with a speculative thread that is not older in program order than the speculative thread from which the load instruction originates).

26. (Previously Amended) A method, comprising:

determining a cache line in a cache corresponding to a speculative thread cache write request includes dirty non-speculative data; and

in response to determining the cache line corresponding to the speculative thread cache write request includes dirty non-speculative data:

generating a writeback of the dirty non-speculative data; and

marking the cache line as speculative.

27. (Previously Amended) The method of claim 26, further comprising: forwarding speculative data from a cache to the speculative thread responsive to a speculative thread cache read request.

28. (Previously Amended) The method of claim 26, further comprising: forwarding non-speculative store data from a cache to the speculative thread responsive to a speculative thread cache read request.

29. (Previously Amended) The method of claim 26, further comprising: forwarding non-speculative data from a cache to a non-speculative thread responsive to a non-speculative thread cache read request.

30. (Previously Amended) The method of claim 26, further comprising processing a cache access request from a non-speculative thread, wherein processing a cache access request from a non-speculative thread comprises:

if a cache does not include a cache line associated with the cache access request, allocating a new cache line;

wherein allocating a new cache line further comprises:

if the new cache line includes dirty speculative data, allocating the new cache line without generating a writeback operation; and

if the new cache line includes dirty non-speculative data, generating a writeback operation.

31. (Previously Amended) The method of claim 26, further comprising: allowing the speculative thread to write data to the cache if the cache line corresponding to the speculative thread cache write request includes speculative data.

32. (Cancelled)

33. (Previously Amended) The method of claim 26, further comprising:

if the cache does not contain data in a cache line corresponding to the speculative thread cache write request:

allocating a new cache line;

marking the new cache line as speculative; and

allowing the speculative thread to write speculative data to the new cache line.

34. (Previously Added) An apparatus comprising:

a processor including:

a first logical processor to execute a speculative thread;

a second logical processor to execute a non-speculative thread;

a storage area to include a speculation identifier (ID) field, the speculation ID field to hold a first value to indicate an associated store instruction is associated with the speculative thread; and

control logic to prevent data associated with the store instruction from being consumed by the non-speculative thread, based on the speculation ID holding the first value.

35. (Previously Added) The apparatus of claim 34, wherein: the first and the second logical processors are the same logical processor, and wherein the non-speculative thread and the speculative thread are to be time multiplexed for execution on the same logical processor.

36. (Previously Added) The apparatus of claim 34, wherein: the storage area includes a store buffer, and wherein the speculation ID field is included within a store buffer entry of the store buffer, the store buffer entry to also hold a first address associated with the store instruction and the data associated with the store instruction.

37. (Previously Added) The apparatus of claim 36, wherein: the first value is to include a first identifier (ID) value associated with the first logical processor.

38. (Previously Added) The apparatus of claim 37, wherein: the control logic includes

comparison logic to compare a second address and a second ID value, which are associated with a load instruction that is to be executed as part of the non-speculative thread on the second logical processor, with the first address and the first ID value; and

store blocker logic to prevent data associated with the store instruction from being consumed by the load instruction that is to be executed as part of the non-speculative thread, in response to the first ID value and the first address not matching the second ID value and the second address.

39. (Previously Added) The apparatus of claim 34, wherein: the processor further includes a third logical processor to execute an additional speculative thread, and wherein the control logic is to allow data associated with the store instruction from being consumed by the additional speculative thread.

40. (Previously Added) The apparatus of claim 34, wherein: the first value is to include a 1-bit mode value.

comparison logic.

41. (Previously Added) The apparatus of claim 34, wherein: the processor further includes marking logic to set the speculation ID field to the first value in response to detecting the store instruction associated with the speculative thread.